Project Report: Student Report Card System

Submitted by: ishan25bai10689-hub

Subject: Python Programming Assignment-1

Date: [23/11/2025]

1. Abstract

The "Student Report Card System" is a Python-based application designed to automate the process of calculating student grades. By taking raw marks as input, the system computes averages, assigns grades based on predefined criteria, and permanently stores the data in a text file. This project demonstrates core programming concepts such as conditional logic, error handling, and file input/output operations.

2. Introduction

2.1 Problem Statement

Manual calculation of student grades for a large class is time-consuming and prone to human error. Additionally, keeping a physical log of these grades can be disorganized.

2.2 Proposed Solution

This project provides a digital solution that:

1. Accepts student details and marks for three subjects.
2. Automatically calculates the total and average marks.
3. Assigns a grade (A, B, C, D, or Fail) using conditional logic.
4. Saves the records to a `class_records.txt` file for future retrieval.

3. System Analysis & Design

3.1 Flow of Logic

The program follows a linear flow with a continuous loop:

1. **Input:** The user provides the Student Name, Registration Number, and Marks for 3 subjects.
2. **Validation:** The system checks if the marks entered are valid integers. If not, it catches the error and prevents a crash.
3. **Processing:** * `Total = Sub1 + Sub2 + Sub3`
   - `Average = Total / 3`
   - `Grade` is determined via an `if-elif-else` ladder.
4. **Output:** The report card is printed to the console.
5. **Storage:** The specific record is appended to a text file.

3.2 Grading Criteria

Average Score	Grade Assigned
90 and above	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	Fail

4. Implementation Details

The project is implemented using Python 3. Key technical features include:

- **Functions:** Modular code using `get_grade(average)` to separate logic from input handling.
- **Exception Handling:** Uses `try-except` blocks to handle `ValueError` if a user types text instead of numbers.
- **File I/O:** Uses the `with open("filename", "a")` context manager to safely append data without overwriting existing records.

5. Results and Output

Upon execution, the system prompts the user for inputs. After calculation, it displays a formatted report card on the screen:

```
--- Simple Student Report Card ---
REPORT CARD FOR: [Student Name]
Total Marks: [Total] / 300
Average: [Average]
Final Grade: [Grade]
```

Simultaneously, a background entry is made in `class_records.txt`: `Name,RegNo,Grade`

6. Conclusion

The Student Report Card System successfully automates the grading process. It ensures data accuracy through validation and data persistence through file handling. This project served as a practical implementation of Python's control structures and file management capabilities.